



*Maryland Department of
Budget & Management*

DBM – people and technology...
a partnership for the new millennium

Office of Information Technology

Systems Development Life Cycle (SDLC)

Volume 3 Glossary and Acronyms

July 2002

45 Calvert Street, Annapolis MD 21401-1907
Tel: (410) 260-7761 • Fax: (410) 974-5045 • Toll Free: (800) 705-3493 • TTY Users: (800) 735-2258
<http://www.dbm.state.md.us>

Table of Contents

Glossary.....	1
Acronyms	23

GLOSSARY

-A-

Acceptance Test - Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. See *User Acceptance Test*.

Accreditation - Formal declaration by an accrediting authority that a computer system is approved to operate in a particular security mode using a prescribed set of safeguards.

Acquisition Plan - A formal document showing how all hardware, software, and telecommunications capabilities, along with resources, are to be obtained during the life of the project.

Activity - A unit of work to be completed in order to achieve the objectives of a work breakdown structure. See *Work Breakdown Structure*. In process modeling, an activity requires inputs and produces outputs. See *Input/Output*.

Adaptability - The ease with which software satisfies differing system constraints and user needs.

Adaptive Maintenance - Maintenance performed to change a system in order to keep it usable in a changed environment.

Alias - A name of a data entity or data attribute that is different from its official name.

Allocated Baseline - The approved documentation that describes the design of the functional and interface characteristics that are allocated from a higher-level configuration item. See *Baseline*.

Alternative Work Patterns - Work pattern that permits tailoring a project plan to meet the specific needs of the project and still conform to SDLC standards.

Application - A system providing a set of services to solve some specific user problem.

Application Model - A model used to graphically and textually represent the required data and processes within the scope of the application development project.

Application Software - Software specifically developed to perform a specific type of work; for example, a word processor. Compare to *System Software*.

Architecture - The structure of a computer system, either a part or the entire system; can be hardware, software, or both.

Audit - A formal review of a project (or project activity) for the purpose of assessing compliance with contractual obligations.

Availability - The degree to which a system (or system component) is operational and accessible when required for use.

-B-

Backup - *v.* To copy software files onto a different media that can be sorted separately from the original files and used to restore the original files, if needed. The act of creating these files. *n.* The set of copied files.

Baseline - A work product (such as software or documentation) that has been formally reviewed, approved, and delivered and can only be changed through formal change control procedures. See *Allocated Baseline, Functional Baseline, Operational Baseline, Product Baseline*.

Benchmark - A standard against which measurements or comparisons can be made.

Bottom-up - The process of designing a system by designing the low-level components first; then integrating them into large subsystems until the complete system is designed; bottom-up testing tests these low-level components first, using software drivers to simulate the higher level components. See *Top-down*.

Build - An operational version of a software product incorporating a specified subset of the complete system functionality. See *Version*.

Business Process Reengineering - The redesign of an organization, culture, and business processes to achieve significant improvements in costs, time, service, and quality.

-C-

Capability - A measure of the expected use of a system.

Capacity - A measure of the amount of input a system could process and/or amount of work a system can perform; for example, number of users, number of reports to be generated.

Certification - Comprehensive analysis of the technical and non-technical security features and other safeguards of a system to establish the extent to which a particular system meets a set of specified security requirements.

Change - In Configuration Management, a formally recognized revision to a specified and documented requirement. See *Change Control, Change Directive, Change Impact Assessment, Change Implementation Notice*.

Change Control – In Configuration Management, the process by which a change is proposed, evaluated, approved (or disapproved), scheduled, and tracked. See *Change, Change Directive, Change Impact Assessment, Change Implementation Notice*.

Change Control Documents - Formal documents used in the configuration management process to track, control, and manage the change of configuration items over the systems development or maintenance life cycle. See *System Change Request, Change Impact Assessment, Change Directive, and Change Implementation Notice*.

Change Directive - The formal Change Control Document used to implement an approved change. See *Change Control Documents*.

Change Impact Assessment - The formal Change Control Document used to determine the effect of a proposed change before a decision is made to implement it. See *Change Control Documents*.

Change Implementation Notice - The formal Change Control Document used to report the actual implementation of a change in a system. See *Change Control Documents*.

Client/Server - A network application in which the end-user interaction with the system (server) is through a workstation (client) that executes some portion of the application.

Code - *v.* To transform the system logic and data from design specifications into a programming language. *n.* The computer program itself; pseudo-code is code written in an English-like logical representation, source code is code written in a programming language, object code is code written in machine language.

Compatibility - A measure of the ability of two or more systems (or system components) to exchange information and use the information that has been exchanged. Same as *Interoperability*.

Component - General term for a part of a software system (hardware or software). See *Product*.

Computer-aided Software Engineering - An electronic tool that is used to assist in the design, development, and coding of software. See *Tools*.

Computer System Security Officer - The person who ensures that all Computer and Telecommunications Security (C&TS) activities are undertaken at the user site. Includes security activities for planning; awareness training; risk management; configuration management; certification and accreditation; compliance assurance; incident reporting; and guidance and procedures.

Concept of Operations - A formal document that describes the user's environment and process relative to a new or modified system; defines the users, if not already known. Called a CONOPS.

Configuration - The functional and/or physical collection of hardware and software components as set forth in formal documentation. Also, the requirements, design, and implementation that define a particular version of a system (or system component). See *Configuration Control*, *Configuration Item*, *Configuration Management*, *Configuration Management Plan*, *Configuration Status Accounting*.

Configuration Audit - Formal review of a project for the purpose of assessing compliance with the Configuration Management Plan.

Configuration Control - The process of evaluating, approving or (disapproving), and coordinating changes to hardware/software configuration items.

Configuration Control Board - The formal entity charged with the responsibility of evaluating, approving (or disapproving), and coordinating changes to hardware/software configuration items.

Configuration Item - An aggregation of hardware and/or software that satisfy an end-use function and is designated by the customer for configuration management; treated as a single entity in the configuration management process. A component of a system requiring control over its development throughout the life cycle of the system.

Configuration Management - The discipline of identifying the configuration of a hardware/software system at each life cycle phase for the purpose of controlling changes to the configuration and maintaining the integrity and trace ability of the configuration through the entire life cycle.

Configuration Management Plan - A formal document that establishes formal configuration management practices in a systems development/maintenance project. See *Configuration Management*.

Configuration Status Accounting - The recording and reporting of the information that is needed to effectively manage a configuration; including a listing of the approved configuration identification, status of proposed changes to the configuration, and the implementation status of approved changes. See *Configuration*.

Contingency Plan - A formal document that establishes continuity of operations processes in case of a disaster. Includes names of responsible parties to be contacted, data to be restored, and location of such data.

Conversion - The process of converting (or exchanging) data from an existing system to another hardware or software environment.

Conversion Plan - A formal document that describes the strategies involved in converting data from an existing system to another hardware or software environment.

Corrective Maintenance - Maintenance performed to correct faults in hardware or software.

Correctness - The degree to which a system or component is free from faults in its specification, design, and implementation.

Cost Analysis - Presents the costs for design, development, installation, operation and maintenance, and consumables for the system to be developed.

Cost-Benefit Analysis - The comparison of alternative courses of action, or alternative technical solutions, for the purpose of determining which alternative would realize the greatest cost benefit; cost-benefit analysis is also used to determine if the system development or maintenance costs still yield a benefit or if the effort should stop.

Cost Estimate - the process of determining the total cost associated with a software development or maintenance project, to include the effort, time, and labor required.

Criteria - A standard on which a decision or judgement may be based; for example, acceptance criteria to determine whether or not to accept a system.

Critical Path - Used in project planning; the sequence of activities (or tasks) that must be completed on time to keep the entire project on schedule; therefore, the time to complete a project is the sum of the time to complete the activities on the critical path.

Critical Review Board - A formal board that guides and monitors the development of requirements that affect current and future State systems.

Customer - An individual or organization who specifies the requirements for and formally accepts delivery of a new or modified system; one who pays for the system. The customer may or may not be the user; see *User*.

-D-

Data Dictionary - A repository of information about data, such as its meaning, relationships to other data, origin, usage and format. A data dictionary manages data categories such as aliases, data elements, data records, data structure, data store, data models, data flows, data relationships, processes, functions, dynamics, size, frequency, resource consumption and other user-defined attributes.

Database Administrator - The person responsible for managing data at a logical level, namely data definitions, data policies and data security.

Database - A collection of logically related data stored together in one or more computerized files; an electronic repository of information accessible via a query language interface.

Database Management System - A software system that controls storing, combining, updating, retrieving, and displaying data records.

Data Store - A repository of data; a file.

Demonstration - A procedure to verify system requirements that cannot be tested otherwise.

Deliverable - A formal product that must be delivered to (and approved by) the customer; called out in the Task Order.

Delivered System Documentation - Includes the Software Development Document, User Manual, Maintenance Manual, and Operations Manual.

Design Phase - The period of time in the systems development life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy system requirements.

Development Phase - The period of time in the systems development life cycle to convert the deliverables of the Design Phase into a complete system.

Disposition Phase - The time when a system has been declared surplus and/or obsolete and the task performed is either eliminated or transferred to other systems.

Disposition Plan - A formal plan providing the full set of procedures necessary to end the operation of the system in a planned, orderly manner and to ensure that system components and data are properly archived or incorporated into other systems.

Document - Written and/or graphical information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, reviews, or results. See *Product*.

-E-

Effectiveness - The degree to which a system's features and capabilities meet the user's needs.

Efficiency - The degree to which a system or component performs its designated functions with minimum consumption of resources.

Element - A subsystem, component, or unit; either software or hardware, as defined by the project.

Enhancement - Maintenance performed to provide additional functional or performance requirements.

Entity - Represents persons, places, events, things, or abstractions that are relevant to the State and about which data are collected and maintained.

-F-

Fault Tolerance - The ability of a system (or system component) to continue normal operation despite the presence of hardware or software faults.

Feasibility - The extent to which the benefits of a new or enhanced system will exceed the total costs and also satisfies the business requirements.

Feasibility Study - A formal study to determine the feasibility of a proposed system (new or enhanced) in order to make a recommendation to proceed or to propose alternative solutions.

Field Test - Testing that is performed at the user site.

Fielded System - An operational system that is installed at the user site.

Full Sequential - The systems development work pattern defined by the nine life cycle phases described in the SDLC Guidance Document.

Functionality - The relative usefulness of a functional requirement as it satisfies a business need.

Functional Baseline - The approved documentation that describes the functional characteristics of the system, subsystem, or component. See *Baseline*.

Functional Configuration Audit - An audit to ensure that the functional requirements have been met by the delivered configuration item. See *Audit*.

Functional Requirement - A requirement that specifies a function (activity or behavior, based on a business requirement) that the system (or system component) must be capable of performing.

Functional Requirements Document - A formal document of the business (functional) requirements of a system; the baseline for system validation.

Functional Test - Testing that ignores the internal mechanism of a system (or system component) and focuses solely on the outputs generated in response to selected inputs and execution conditions. Same as black box testing.

-G-

Gantt Chart - A list of activities plotted against time, showing start time, duration, and end time; also known as a bar chart.

-H-

Hardware - The physical portion of a system (or subsystem), including the electrical components. Compare to *Software*.

Host - The computer that controls communications in a network that administers a database; the computer on which a program or file is installed; a computer used to develop software intended for another computer. See *Target*.

-I-

Implementation - Installing and testing the final system, usually at the user (field) site; the process of installing the system.

Implementation Phase - The period of time in the systems development life cycle when the system is installed, made operational, and turned over to the user (for the beginning of the Operations and Maintenance Phase).

Implementation Plan - A formal document that describes how the system will be installed and made operational.

Information Technology - The application of engineering solutions in order to develop computer systems that process data.

In-Process Review - Formal review conducted (usually annually) during the Operations and Maintenance Phase to evaluate system performance, user satisfaction with the system, adaptability to changing business needs, and new technologies that might improve the system.

In-Process Review Report - A formal document detailing the findings of the In-Process Review. See *In-Process Review*.

Input/Output - The process of entering information into a system (input) and its subsequent results (output). A hardware device that enables input (for example, a keyboard or card reader) and output (for example, a monitor or printer). Collectively known as I/O.

Inspection - A semiformal to formal technique in which software requirements, design, or code are examined in detail by a person or group other than the originator to detect errors. See *Peer Review*, *Walk-through*.

Integrated Product Team - A multidisciplinary group of people who support the Project Manager in the planning, execution, delivery and implementation of life cycle decisions for the project.

Integration Document - A formal document that describes how the software components, hardware components, or both are combined and the interaction between them.

Integration and Test Phase - Life cycle phase during which subsystem integration, system, security, and user acceptance testing are conducted; done prior to the Implementation Phase.

Integration Test - Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them.

Integrity - The degree to which a system (or system component) prevents unauthorized access to, or modification of, computer programs or data.

Iterative - A procedure in which repetition of a sequence of activities yields results successively close to the desired state; for example, an iterative life cycle in which two or more phases are repeated until the desired product is developed.

Interface - To interact or communicate with another system (or system component). An interface can be software and/or hardware. See *User Interface*.

Interface Control Document - Specifies the interface between a system and an external system(s).

Interoperability - A measure of the ability of two or more systems (or system components) to exchange information and use the information that has been exchanged. Same as *Compatibility*.

Information Technology Systems Security Certification and Accreditation - A formal set of documents showing that the installed security safeguards for a system are adequate and work effectively.

-L-

Lessons Learned - A formal or informal set of examples collected from experience (for example, experience in system development) to be used as input for future projects to know what went well and what did not; collected to assist other projects.

Library - A configuration controlled repository for system components (for example, documents and software).

Life Cycle - All the steps or phases a project passes through during its system life; from concept development to disposition. There are nine life cycle phases in the SDLC.

-M-

Maintainability - The ease with which a software system (or system component) can be modified to correct faults, improves performance, or other attributes, or adapt to a changed environment.

Maintenance - In software engineering, the activities required to keep a software system operational after implementation. See *Adaptive Maintenance*, *Corrective Maintenance*, *Enhancement*, *Perfective Maintenance*.

Maintenance Manual - A formal document that provides systems maintenance personnel with the information necessary to maintain the system effectively.

Maintenance Review - A formal review of both the completed and pending changes to a system with respect to the benefits achieved by completing the recommended changes; also provides information about the amount of maintenance required based on activity to date. Part of the *Post-Implementation Review Report*.

Measurement - In project management, the process of collecting, analyzing and reporting metrics data.

Methodology - A set of methods, procedures, and standards that define the approach for completing a system development or maintenance project.

Metrics - A quantitative measure of the degree to which a system, component, or process possesses a given attribute.

Migration - Porting a system, subsystem, or system component to a different hardware platform.

Milestone - In project management, a scheduled event that is used to measure progress against a project schedule and budget.

Mission - The goals or objectives of an organization or activity.

Model - A simplified representation or abstraction (for example, of a process, activity, or system) intended to explain its behavior.

Module - In system design, a software unit that is a logically separate part of the entire program. See *Unit*.

-N-

Non-technical - Relating to agreements, conditions, and/or requirements affecting the management activities of a project. Compare to *Technical*.

-O-

Operational Baseline - Identifies the system accepted by the users in the operational environment after a period of onsite test using production data. See *Baseline*.

Operations Manual - A formal document that provides a detailed operational description of the system and its interfaces.

Operations and Maintenance (O&M) Phase - The period of time in the systems development life cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements.

-P-

Peer Review - A formal review where a product is examined in detail by a person or group other than the originator. See *Inspection*, *Walk-through*.

Perfective Maintenance - Software maintenance performed to improve the performance, maintainability, or other attributes of a computer program.

Performance Measures - A category of quality measures that address how well a system functions.

Performance Measurement and Capacity Planning - A set of procedures to measure and manage the capacity and performance of information systems equipment and software.

Performance Review - Formal review conducted to evaluate the compliance of a system or component with specified performance requirements.

Phase - A defined stage in the systems development life cycle; there are nine phases in the full, sequential life cycle.

Phase Review - A formal review conducted during a life cycle phase; usually at the end of the phase or at the completion of a significant activity.

Physical Configuration Audit - An audit to ensure that all physical attributes listed in the design requirements have been met by the configuration item being delivered.

Pilot - An alternative work pattern to develop a system to demonstrate that the concept is feasible in an operational environment. Pilots are used to provide feedback to refine the final version of the product and are fielded for a preset, limited period of time. Compare to a *Prototype*.

Planning Phase - The period of time in the systems development life cycle in which a comprehensive plan for the recommended approach to the systems development or maintenance project is created. Follows the Systems Concept Development Phase, in which the recommended approach is selected.

Post-Implementation Review - A formal review to evaluate the effectiveness of the systems development effort after the system is operational (usually for at least six months).

Post-Implementation Review Report - A formal document detailing the findings of the Post-Implementation Review. See *Post-Implementation Review*.

Post-Termination Review - A formal review to evaluate the effectiveness of a system disposition.

Post-Termination Review Report - A formal document detailing the findings of the Post-Termination Review. See *Post-Termination Review*.

Procedure - A series of steps (or instructions) required to perform an activity. Defines “how” to perform an activity. Compare to *Process*.

Process - A finite series of activities as defined by its inputs, outputs, controls (for example, policy and standards), and resources needed to complete the activity. Defines “what” needs to be done. Compare to *Procedure*.

Process Model - A graphical representation of a process.

Process Review - A formal review of the effectiveness of a process.

Product - General term for an item produced as the result of a process; can be a system, subsystem, software component, or a document.

Product Baseline - The set of completed and accepted system components and the corresponding documentation that identifies these products. See *Baseline*.

Production - A fully documented system, built according to the SDLC, fully tested, with full functionality, accompanied by training and training materials, and with no restrictions on its distribution or duration of use.

Product Review - A formal review of a product software (or document) to determine if it meets its requirements. Can be conducted as a *peer review*.

Program Specification - A description of the design logic in a software component, generally using pseudo-code. See *Code*.

Project - The complete set of activities associated with all life cycle phases needed to complete a systems development or maintenance effort from start to finish (may include hardware, software, and other components); the collective name for this set of activities. Typically a project has its own funding, cost accounting, and delivery schedule.

Project Management - The process of planning, organizing, staffing, directing, and controlling the development and/or maintenance of a system.

Project Management Plan - A formal document detailing the project scope, activities, schedule, resources, and security issues. The Project Management Plan is created during the Planning Phase and updated through the Disposition Phase.

Project Manager - The person with the overall responsibility and authority for the day-to-day activities associated with a project.

Prototype - A system development methodology to evaluate the design, performance, and production potential of a system concept (it is not required to exhibit all the properties of the final system). Prototypes are installed in a laboratory setting and not in the field, nor are they available for operational use. Prototypes are maintained only long enough to establish feasibility. Compare to a *Pilot*.

-Q-

Quality - The degree to which a system, component, product, or process meets specified requirements.

Quality Assurance - A discipline used by project management to objectively monitor, control, and gain visibility into the development or maintenance process.

Quality Assurance Plan - A formal plan to ensure that delivered products satisfy contractual agreements, meet or exceed quality standards, and comply with approved systems development or maintenance processes.

Quality Assurance Review - A formal review to ensure that the appropriate Quality Assurance activities have been successfully completed, held when a system is ready for implementation.

-R-

Rapid Application Development - In a RAD work pattern, the Requirements Definition and Design phases are iteratively conducted; in this process, a rough set of requirements is used to create an initial version of the system, giving users visibility into the look, feel, and system capabilities. User evaluation and feedback provide revisions to the requirements, and the process is repeated until the requirements are considered to be complete.

Records Management - The formal set of system records (for example, files, data) that must be retained during the Disposition Phase; the plan for collecting and storing these records.

Recoverability - The ability of a software system to continue operating despite the presence of errors.

Reengineering - Rebuilding a process to suit some new purpose; for example, a new business process.

Regression Test - In software maintenance, the rerunning of test cases that previously executed correctly in order to detect errors introduced by the maintenance activity.

Release - A configuration management activity wherein a specific version of software is made available for use.

Reliability - The ability of a system (or system component) to perform its required functions under stated conditions for a specified period of time.

Requirement - A capability needed by a user; a condition or capability that must be met or possessed by a system (or system component) to satisfy a contract, standard, specification, or other formally imposed documents.

Requirements Analysis Phase - The period of time in the systems development life cycle during which the requirements for a software product are formally defined, documented and analyzed.

Requirements Management - Establishes and controls the scope of system development efforts and facilitates a common understanding of system capabilities between the System Proponent, developers, and future users.

Requirements Trace ability Matrix - Provides a method for tracking the functional requirements and their implementation through the development process.

Resource - In management, the time, staff, capital and money available to perform a service or build a product; also, an asset needed by a process step to be performed.

Reverse Engineering - A software engineering approach that derives the design and requirements of a system from its code; often used during the maintenance phase of a system with no formal documentation.

Review - A formal process at which an activity or product (for example, code, document) is presented for comment and approval; reviews are conducted for different purposes, such as peer reviews, user reviews, management reviews (usually for approval) or done at a specific milestone, such as phase reviews (usually to report progress).

Review Report - A formal document that records the results of a review.

Risk - A potential occurrence that would be detrimental to the project; risk is both the likelihood of the occurrence and the consequence of the occurrence.

Risk Assessment - The process of identifying areas of risk; the probability of the risk occurring, and the seriousness of its occurrence; also called risk analysis.

Risk Management - The integration of risk assessment and risk reduction in order to optimize the probability of success (that is, minimize the risk).

Risk Management Plan - A formal document that identifies project risks and specifies the plans to reduce these risks.

Role - A defined responsibility (usually task) to be carried out by one or more individuals.

-S-

Scope - The established boundary (or extent) of what must be accomplished; during planning, this defines what the project will consist of (and just as important, what the project will not consist of).

Security - The establishment and application of safeguards to protect data, software, and hardware from accidental or malicious modification, destruction, or disclosure.

Security Risk Assessment - Tool that permits developers to make informed decisions relating to the acceptance of identified risk exposure levels or implementation of cost-effective measures to reduce those risks. See *Requirements Analysis Phase*.

Security Test - A formal test performed on an operational system, based on the results of the security risk assessment in order to evaluate compliance with security and data integrity guidelines, and address security backup, recovery, and audit trails. Also called Security Testing and Evaluation (ST&E).

Segment - A major part of a larger system or subsystem; in software, a self-contained portion of a computer program.

Sensitive System - A system or subsystem that requires an IT Systems Security Certification and Accreditation; contains data requiring security safeguards.

Sensitivity Analysis - Assesses the potential effect on inputs (costs) and outcomes (benefits) that depends on the relative magnitude of change in certain factors or assumptions.

Software - Computer programs (code), procedures, documentation, and data pertaining to the operation of a computer system. Compare to *Hardware*.

Software Development Document - Contains all of the information pertaining to the development of each unit or module, including the test cases, software, test results, approvals, and any other items that will help explain the functionality of the software.

Standard - Mandatory requirements to prescribe a disciplined uniform approach to software development and maintenance activities.

Subsystem - A collection of components that meets the definition of a system, but is considered part of a larger system. See *System*.

Survivability - A measure of the ability of a system to continue to function, especially in the presence of errors.

System - A collection of components (hardware, software, interfaces) organized to accomplish a specific function or set of functions; generally considered to be a self-sufficient item in its intended operational use.

System Administrator - The person responsible for planning a system installation and use of the system by other users.

System Boundary Document - A formal document created during the System Concept Development Phase which lists the business case for initiating the system or project. It contains the mission, goals, objectives, high level requirements, critical success factors, assumptions, responsible persons, projected costs associated with the investment, risks, scope, schedule, milestones, etc.

System Component - Any of the discrete items that comprise a system or subsystem. See *Subsystem*, *System*.

System Change Request - The formal Change Control Document procedure used to request a change to a system baseline, provide information concerning the requested change, and act as the documented approval mechanism for the change. See *Change Control Documents*.

System Concept Development Phase - Phase that begins after the need or opportunity has been identified in the Initiation Phase. The approaches for meeting this need are reviewed for feasibility and appropriateness (for example, cost-benefit analysis) and documented in the SBD.

System Design Document - A formal document that describes the system architecture, file and database design, interfaces, and detailed hardware/software design; used as the baseline for system development.

System Proponent - The organization benefiting from or requesting the project; frequently thought of as the “customer” for that project.

Systems Administration Manual - A manual that serves the purpose of an Operations Manual in a distributed (client/server) application. See *Operations Manual*, *Client/Server*.

Systems Analysis - In systems development, the process of studying and understanding the requirements (customer needs) for a system in order to develop a feasible design.

Systems Development Life Cycle - A formal model of a hardware/software project that depicts the relationship among activities, products, reviews, approvals, and resources. Also, the period of time that begins when a need is identified (initiation) and ends when the system is no longer available for use (disposition).

Systems Manager - The individual, or group, responsible for post-implementation system maintenance, configuration management, change control, and release control. This may or may not include members of the development team.

System Security Plan - A formal document that establishes the processes and procedures for identifying all areas where security could be compromised within the system (or subsystem).

System Software - Software designed to facilitate the operation of a computer system and associated computer programs (for example, operating systems, code compilers, utilities). Compare to *Application Software*.

System Test - The process of testing an integrated hardware/software system to verify that the system meets its documented requirements.

-T-

Tailor - A formal procedure to modify a process, standard, procedure, or work pattern to fit a specific use or business need.

Target - The computer that is the destination for a host communication; See *Host*. In programming, a language into which another language is to be translated.

Task - In project management, the smallest unit of work subject to management accountability; a work assignment for one or more project members fulfilling a role, as defined in a work breakdown structure.

Technical - Relating to agreements, conditions, and/or requirements affecting the functionality and operation of a system. Compare to *non-technical*.

Test - The process of exercising the product to identify differences between expected and actual results and performance. Typically testing is bottom-up: unit test, integration test, system test, and acceptance test.

Test Case - A specific set of test data and associated procedures developed for a particular test.

Test Files/Data - Files/data developed for the purpose of executing a test; becomes part of a test case. See *Test Case*.

Testability - A metric used to measure the characteristics of a requirement that enable it to be verified during a test.

Test Analysis Approval Determination - The form attached to the Test Analysis Report as a final result of the test reviews for all testing levels above the Integration test. See *Test Analysis Report*.

Test Analysis Report - Formal documentation of the software testing as defined in the Test Plan.

Test and Evaluation (T&E) - T&E occurs during all major phases of the development life cycle, beginning with system planning and continuing through the operations and maintenance phase, ensures standardized identification, refinement, and trace ability of the requirements as such requirements are allocated to the system components.

Test and Evaluation Master Plan - The formal document that identifies the tasks and activities so the entire system can be adequately tested to assure a successful implementation.

Test Problem Report - Formal documentation of problems encountered during testing; the form is attached to the Test Analysis Report. See *Test Analysis Report*.

Test Readiness Review - A formal phase review to determine that the test procedures are complete and to ensure that the system is ready for formal testing.

Tools - Software application products that assist in the design, development, and coding of software. Also called CASE tools; see *Computer-aided Software Engineering*.

Top-down - An approach that takes the highest level of a hierarchy and proceeds through progressively lower levels; compare to *Bottom-up*.

Trace ability - In requirements management, the identification and documentation of the derivation path (upward) and allocation path (downward) of requirements in the hierarchy.

Training - The formal process of depicting, simulating, or portraying the operational characteristics of a system or system component in order to make someone proficient in its use.

Training Plan - A formal document that outlines the objectives, needs, strategy, and curriculum to be addressed for training users of the new or enhanced system.

-U-

Unit - the smallest logical entity specified in the design of a software system; must be of sufficient detail to allow the code to be developed and tested independent of other units. See *Module*.

Unit Test - In testing, the process of ensuring that the software unit executes as intended; usually performed by the developer.

Upgrade - A new release of a software system for the purpose of including a new version of one or more system components.

Usability - The capability of the software product to be understood, learned, used and is of value to the user, when used under specified conditions.

User - An individual or organization who operates or interacts directly with the system; one who uses the services of a system. The user may or may not be the customer. See *Customer*.

User Acceptance Test - Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the user to determine whether or not to accept the system. See *Acceptance Test*.

User Interface - The software, input/output (I/O) devices, screens, procedures, and dialogue between the user of the system (people) and the system (or system component) itself. See *Interface*.

User Manual - A formal document that contains all essential information for the user to make full use of the new or upgraded system.

User Satisfaction Review - A formal survey used to gather the data needed to analyze current user satisfaction with the performance capabilities of an existing system or application; administered annually, or as needed.

-V-

Validation - The process of determining the correctness of the final product, system, or system component with respect to the user's requirements. Answers the question, "Am I building the right product?" Compare to *Verification*.

Verifiability - A measure of the relative effort to verify a requirement; a requirement is verifiable only if there is a finite cost-effective process to determine that the software product or system meets the requirement.

Verification - The process of determining whether the products of a life cycle phase fulfill the requirements established during the previous phase; answers the question, “Am I building the product right?” Compare to *Validation*.

Verification and Validation Plan - A formal document that describes the process to verify and validate the requirements. Created during the Planning Phase and updated throughout the SDLC.

Version - An initial release or re-release of a computer software configuration item, associated with a complete compilation or recompilation of the computer software configuration item; sometimes called a build. See *Build*.

Version Description Document - A formal document that describes the exact version of a configuration item and its interim changes. It is used to identify the current version; provides a “packing list” of what is included in the release.

Volatility - In requirements management, the degree to which requirements are expected to change throughout the systems development life cycle; opposite of stability.

-W-

Walk-through - A software inspection process, conducted by peers of the software developer, to evaluate a software component. See *Inspection*, *Peer Review*.

Work Breakdown Structure - In project management, a hierarchical representation of the activities associated with developing a product or executing a process; a list of tasks; often used to develop a *Gantt chart*.

Work Pattern - The complete set of life cycle phases, activities, deliverables, and reviews required to develop or maintain a software product or system; a formal approach to systems development.

ACRONYMS

-A-

ABL	Allocated Baseline
ACSN	Advanced Change Study Notice
ADP	Automated Data Processing
AIS	Automated Information System
APB	Acquisition Project Baseline

-B-

BPR	Business Process Reengineering
------------	--------------------------------

-C-

CASE	Computer-Aided Software Engineering
CBA	Cost-Benefit Analysis
CD-ROM	Compact Disk-Read Only Memory
CI	Configuration Item
CIO	Chief Information Officer
CM	Configuration Management
CMM	Capability Maturity Model
COCOMO	Constructive Cost Model
CONOPS	Concept of Operations
COTR	Contracting Officers' Technical Representative
COTS	Commercial Off-The-Shelf
CRUD	Create, Read, Update, Delete

CSA	Configuration Status Accounting
CSCI	Computer Software Configuration Items
CWBS	Contract Work Breakdown Structure

-D-

DFD	Date Flow Diagrams
DBA	Database Administrator
DBMS	Database Management System

-E-

EA	Enterprise Architecture
ECP	Engineering Change Proposal

-F-

FBL	Functional Baseline
FCA	Functional Configuration Audit
FIPS	Federal Information Processing Standards
FRD	Functional Requirements Document

-G-

GUI	Graphical User Interface
------------	--------------------------

-H-

HCI	Hardware Configuration Items
------------	------------------------------

-I-

ICD	Interface Control Document
IEEE/EIA	Institute of Electrical and Electronics Engineers/Electronic Industries Assoc.

IPT	Integrated Product Team
IRM	Information Resources Management
ISO	International Standard Organization
IT	Information Technology
ITIB	Information Technology Investment Board

-J-

JCL	Job Control Language
------------	----------------------

-L-

LAN	Local Area Network
------------	--------------------

-O-

-P-

PBL	Product Baseline
PCA	Physical Configuration Audit
PERT	Program Evaluation Review Technique

-Q-

QA	Quality Assurance
QAR	Quality Action Report

-R-

RFP	Request for Proposal
RM	Risk Management
RTM	Requirements Trace ability Matrix

-S-

SBD	System Boundary Document
SBU	Sensitive-but-Unclassified
SCR	System Change Request
SDLC	Systems Development Life Cycle
SEI	Software Engineering Institute
SEMP	Systems Engineering Management Plan
SLIM	Software Life Cycle Management
SOW	Statement of Work
SPM	Systems Program Manager
ST&E	Security Test and Evaluation

-T-

TAAD	Test Analysis Approval Determination
T&E	Test and Evaluation
TIM	Technical Interchange Meetings
TPM	Technical Performance Measurement
TRM	Technical Reference Model

-V-

VDD	Version Description Document
V&V	Verification and Validation

-W-

WBS	Work Breakdown Structure
------------	--------------------------